Discussion Paper - JSON Encodings for EO Coverages

This discussion paper was submitted to the ESDIS Standards Office (ESO) by the NASA Earth Science Data System Working Group focused on exploring JSON Encodings in Earth Observation Coverages.

The following questions are meant to facilitate discussion. You may respond by commenting on them or on any other part of the document.

1. Are you using or experimenting with any of the EO JSON coverage formats mentioned in Section 3? (CoverageJSON/CovJSON, HDF5/JSON, NCO-JSON)
2. Are there other JSON formats you are using for encoding coverage data (or other geographic data)?
3. For the previous questions, if you answered "yes":
    1. What are you using any of these formats for?
    2. Do these formats meet your needs?
    3. Do you recommend wider use of any of these formats?
    4. Are you using any of these formats to replace other formats or are you using them in addition to other formats?
4. In your experience so far, is JSON appropriate as a container for EO coverage data (see section 3.3)?
5. Are there any findings in this paper that you don't agree with?
6. Are there topics that you think are important in this context but not covered in this paper?
7. Are there additional resources you would suggest people should know about?
8. What value, if any, would harmonization between efforts (e.g. NCOJSON and CF-JSON) as detailed within this discussion paper have on the wider community?
9. Is it worth formalizing a set of 'best practices' as covered in Subsection 3.3 The Appropriateness of JSON as a Container for EO Ranges of this discussion paper?
10. If you are familiar with the OGC Testbed 14 report on Swath Coverages (HTML http://docs.opengeospatial.org/per/18-047r3.html, PDF http://docs.opengeospatial.org/per/18-047r3.pdf): Seeing as the goal of this task was to improve access to NASA data in swath structure, specifically for those Earth Observations in Level 1 and 2, could JSON Encodings for Earth Observation Coverages be used to compliment the research output included within this report?

Please provide any feedback you might have to this document and contribute to the discussion by adding or responding to comments.

To add a comment, select a word or words, then click the "insert comment" button above, it looks like this: 

To respond to an existing comment, click on the text in the comment and enter your reply in the box that opens up just under the comment.

Abstract

This discussion paper documents and concludes one year (2018-2019) of work undertaken by a NASA Earth Science Data System Working Group focused on exploring *JSON Encodings in Earth Observation Coverages.* The primary function of this paper is to ensure that the collective Working Group knowledge obtained from the year effort is not lost and consequently that it can be considered, debated and hopefully utilized in other forums outside of NASA with the aim of driving progress in this field.

This discussion paper will be of particular interest to the following parties

- **Web application developers;** tasked with designing and developing applications which consume Earth Observation spatial data encoded as JSON.
- **Parties (including standards bodies) interested in serving and consuming Spatial data on the Web** e.g. W3C, OGC or developers of other data standards, etc.

This is a discussion paper: we invite readers to make comments using the Google Docs comment feature. Some questions we are interested in seeing discussed are:

1. Seeing as the goal of the Swath Coverage task (as detailed within the OGC Testbed-14: Swath Coverage Engineering Report) was to improve access to National Aeronautics and Space Administration (NASA) data in Swath structure, specifically for those Earth Observations in Level 1 and 2. Could JSON Encodings for Earth Observation Coverages be used to compliment the research output included within this report?
2. What value, if any, would harmonization between efforts (e.g. NCOJSON and CF-JSON) as detailed within this discussion paper have on the wider community?
3. To what extent is standardization of JSON Encodings a factor in their uptake and adoption?
4. Is it worth formalizing a set of ‘best practices’ as covered in *Subsection 3.3 The Appropriateness of JSON as a Container for EO Ranges* of this discussion paper?

Table of Contents

1 Introduction

This discussion paper documents and concludes one year (2018-2019) of work undertaken by a NASA Earth Science Data System Working Group (ESDSWG) [1] focused on exploring *JSON* [2] *Encodings in Earth Observation (EO) Coverages* (henceforth referred to as the '*WG'*). This one year effort was established so that the ESDSWG community (and beyond) could better understand the state of the art in this field. The WG tasked itself to the following mission statement:

> "[The WG will] *Align efforts, reduce overlap and expand/evangelize usage and software implementations demonstrating the use of JSON encodings for representing EO coverages and related implementations. Primary output will be a technical note/discussion paper representing a review of available JSON encodings and implementation status."*

The WG set out to achieve this mission by hosting authors, maintainers and industry experts from a number of candidate JSON encodings to present at monthly WG teleconferences. Additionally, the WG co-organized and co-hosted various sessions at well-attended events such as ESIP's 2018 Summer Meeting [3][4] and the 2018 AGU Fall Meeting [5] which resulted in a wealth of content. It goes without saying that these events resulted in a significant increase in technology awareness within the WG and encouraged collective information sharing and collaboration from attendees.

The remainder of this paper is structured as follows; *Section 2 Background* – provides our rationale for creating the WG from the ESDSWG perspective. *Section 3 Discussion* – facilitating our review of the technology space and the maturity of various technologies within the overall EO ecosystem. We conclude Section 3 with an account of current and ongoing harmonization efforts, which we consider to be a core contribution of this discussion paper.

2 Background

As JSON continues to prosper in its current state of ubiquity, it is no wonder that a rich variety of EO geospatial JSON encodings already exist and are under active development. In fact, NASA's Earth Observing System Data and Information System (EOSDIS) [6] is a consumer and developer of several JSON-related efforts as well as standards development via the Open Geospatial Consortium (OGC) [7] amongst others. In 2018, community members from NASA's

ESDSWG identified a requirement to align efforts, reduce overlap and expand/evangelize usage and software implementations, as related to *JSON Encodings in Earth Observation Coverages,* for the benefit of EOSDIS.

Table 1 (listed alphabetically) shows the WG's early initial attempt at capturing relevant EO-related JSON efforts. The forthcoming Section titled *Technology Review* provides a subset of Table 1 which the WG actually explored.

Table 1: An initial collection of EO-related JSON efforts

Name	Purpose	Curator	Links	Notes
CF-JSON	Coverage data	David Johnson, Met Ocean, New Zealand	[8]	Can express data in any CF compliant NetCDF file See Note 1 below
CovJSON	Coverage data	W3C/OGC	[9]	Coverage data, any CRS, type mirrors OGC Coverage Implementation Specification See Note 1 below.
GeoJSON	Vector data	IETF	[10]	Vector data, WGS84 only, types mirror OGC Simple Features
HDF5/JSON	Coverage data	The HDF Group	[11] [12]	A specification, library, and utilities for describing HDF5 content in JSON
JSON-LD	Linked data	W3C	[13]	Linked data spec, not specifically for EO
NCO-JSON	Coverage data	Charlie Zender	NCO spec section on JSON output via --jsn or --json flags [14] [15] Sourceforge discussion [16] Github specification [17]	See Note 1 below
OGC EO Dataset Metadata GeoJSON(-LD) Encoding Standard	Metadata	OGC	[18] (Both OGC Portal access and view permission required)	Can be applied to encode metadata based on the Earth Observation Metadata Profile of Observations and Measurements (O&M) OGC 10-157r4, or as an encoding of the Unified Metadata Model for Granules (UMM-G) conceptual model
OGC OpenSearch Extension for Earth Observation	Search results	OGC	[19]	WGISS effort to have GeoJSON be a valid OpenSearch result encoding.
OGC OWS Context GeoJSON Encoding Standard	Service context	OGC	[20]	OWS Context allows a set of configured information resources to be passed between applications, primarily as a collection of services.
STAR JSON	Coverage data	Pedro Vicente	[21] [22]	STAR JSON is a JSON schema that is used to share commonly used scientific data formats, such as HDF5 and netCDF

TopoJSON	Vector data	Mike Bostock	[23]	An extension of GeoJSON that encodes topology. Rather than representing geometries discretely, geometries in TopoJSON files are stitched together from shared line segments called *arcs*.

Note 1: Discussion on harmonizing CovJSON, CF-JSON and NCO-JSON can be found at [24]

At this stage, it is important to make some contextual notes

- The WG tasked itself, very intentionally, with reviewing *only* JSON encodings which address the representation of EO coverages. In this context we leverage the definition of a coverage as "… a function from a spatial, temporal or spatiotemporal domain to an attribute range. A coverage associates a position within its domain to a record of values of defined data types." [25]. A specific type of coverage of interest to EOSDIS would be a satellite swath coverage from one of NASA's Earth observing platforms. Such an example is provided in [26]. As it turned out, this distinction was of benefit to the WG activities as it constrained the WG scope considerably resulting in us reducing the number of encodings ultimately reviewed[1]. It is worth noting that the use of *coverages* as a standalone term can still cause some degree of confusion and disagreement between communities.
- The WG mission statement specified that it would "…*Align efforts, reduce overlap [in] …the use of JSON encodings for representing EO coverages…*" however no intentional pressure or influence was dedicated to any particular initiative in any particular way. This is to say that the WG tried its best to ensure non-biased review of each encoding. The aim was to review each based on their individual merit as opposed to approaching the year-long review with any particular bias in mind from the outset. We hope this is evident from the complementary technical evaluation which comprises the bulk of the remainder of the document.
- Building on from point #2 above, the WG found contextual conversation such as that available at [24] to be of the utmost value when piecing together this narrative. Without consulting these informative resources, it will be challenging for readers to gain full understanding of the state of topics such as *alignment of efforts*, *harmonization*, etc. We therefore highly encourage readers to pay close attention to informative resources such as discussion threads on GitHub issues.

3 Discussion

This section focuses primarily on the WG technology evaluation and maturity review. As stated in the contextual notes above, the number of JSON encodings capable of representing coverages that the WG actually reviewed was reduced significantly from those originally identified in Table 1. This resulted in the WG investigating only ***CovJSON*** [9], ***HDF5/JSON*** [11] and ***NCO-JSON*** [14] with any degree of detail. ***This is not to say that other encodings e.g. STAR JSON, are of no use***. The WG activity period simply expired. Commentary on technology review and ecosystem maturity is included in Sections 3.2 and 3.3 respectively. First however,

[1] These are presented in Section 3.

Section 3.1 includes accompanying material the WG managed to review throughout the course of the WG activities.

3.1 Additional Background Material

The following efforts were also brought to the WG attention. As they are JSON-related, readers may find them of interest. These resources however did not directly relate to *JSON encodings for representing EO coverages*, so were deemed outside the scope of the encodings covered in Section 2.2:

- **netCDF-LD** [27]: is an approach for constructing Linked Data descriptions using the metadata and structures found in netCDF files. netCDF-LD enhances netCDF metadata, enabling information found in netCDF files to be linked with published conventions and controlled vocabularies used to express the content.
- **science-on-schema.org** [28]: Provides guidance for publishing schema.org as JSON-LD for the Geosciences.
- **OGC Testbed-12 JSON and GeoJSON User Guide** [29]: This work generally presents the JSON data format. It discusses important questions such as what JSON offers, when a JSON encoding offers an advantage (e.g. in simplicity and flexibility) to an XML encoding, and when XML encodings can provide a better solution (e.g. adding more expressivity and robustness). Finally, it presents some precision that OGC can add on top of the basic JSON definition to ensure better interoperability of applications using JSON. Specifically, however, the subsection on ***JSON for coverages*** provides some historical narrative as follows "…The idea of creating a JSON GMLCov associated to a JSON coverage appears for the first time in the section 9 of the OGC 15-053r1 Testbed-11 Implementing JSON/GeoJSON in an OGC Standard Engineering Report. This idea was taken by the MELODIES FP7 project (http://www.melodiesproject.eu/), and described as a full specification, as well as implemented as an extension of the popular map browser *Leaflet*. The description of the approach can be found here https://github.com/covjson/specification... that is to say it introduces CovJSON as a suitable encoding for EO coverages. The reader is again encouraged to read more extensively into this document as the narrative is developed further through worked examples. The work described in [30] is extended by the OGC Architecture DWG *JSON Best Practices* subgroup, who are compiling JSON-related information into a set of best practices [31]. The text relating to EO Coverages is almost identical in both [30] and [31].

3.2 Technology Review

Table 2 provides additional information (listed alphabetically) on the subset of JSON encodings noted above and communicates a very positive overall spread perspective with regards to uptake, activity, harmonization and standardization.

Table 2: Additional information for three candidate JSON encodings for EO data

Name	Example Implementations	Official Standard Body Support (as of date of publication)	Notes
CovJSON	Hyrax (>=1.15.0), LeafletJS, NASA WorldWind, ncWMS, pycovjson, JavaScript API, CovJSON Playground Web Application, Online 'Cookbook' documentation, Java library implemented as an EDAL module.	**Yes** A wealth of work was undertaken by the joint W3C + OGC Spatial Data on the Web Working Group on providing an overview of CovJSON [32] A formal standardization effort is now being facilitated through the OGC's Coverages DWG [33]	Primarily designed/focused on encoding a wide variety of EO domain objects Reasonably mature specification. Future efforts should focus on a simple, JSON-first standard that is practical for tool developers. This will include the development of a test suite. Other OGC initiatives such as the generation of Web Services based on the WFS3.0 draft standard [34] and the OGC Web OpenAPI guidelines [35] may produce useful synergies. Additional GitHub conversation threads exist for comparing and contrasting CoverageJSON with CF-NetCDF [36] but as of writing this effort seems stagnant.
HDF5/JSON	Highly Scalable Data Server (HSDS), h5json Python package, HDF Product Designer	**No**	Primarily designed/focused on describing actual file content. Although no standards body effort is ongoing, the specification and tooling appear to be reasonably stable. As one would expect, HDF5/JSON does not cater for coverages represented outside of the HDF5 ecosystem so supplementary tooling is required for this encoding to be more widely used. HDF5/JSON tools provide roundtrip conversion between the content of any HDF5 file (netCDF-4 files included) and HDF5/JSON. As the OGC HDF SWG [37] moves towards standardization of the HDF5 data model, there may be an uptake in HDF5/JSON outside of The HDF Group.

NCO-JSON	NCO, ERDDAP, CF-JSON, STAR-JSON, various applications in agriculture (BRAPI, AgMIP/ICASA)	**No**	Primarily design/focused on representing the NetCDF CF metadata conventions Reasonably mature specification with uptake within several other projects. NCO's JSON tooling can create JSON with a one-to-one representation of the information that is in any CF/NetCDF file. As of early 2019, renewed efforts are ongoing to consolidate CF-JSON and NCO-JSON [38].

Specifically, the following items are noteworthy:

- The Hyrax data server v1.15.0 release included CovJSON as a response handler meaning that clients can now request CovJSON natively. This is a significant step forwards for CovJSON as Hyrax is deployed extensively in order to facilitate distribution of EOSDIS data holdings. Additionally, standardization via the OGC Coverages DWG should provide a positive move for CovJSON. Readers that are interested in CovJSON relationships with other data models e.g. HDF5, NetCDF and CF-NetCDF, OGC Coverage Implementation Schema, and OGC TimeseriesML should engage with the *Section 5: Relationship with other data models* of [32].
- Although HDF5/JSON is not a suitable representation for coverages other than those that can be represented in HDF5, with the standardization of HDF5 via the OGC HDF SWG, this may present a renewed opportunity for stakeholders to investigate additional uptake of the encoding.
- NCO-JSON could potentially continue to extend its usefulness and interoperability by engaging in consolidation efforts with CF-JSON as is indicated in [38]. As with CovJSON, it appears that no formal test suite (in the case of NC-JSON this would be against netCDF3/4) exists for formally validating NCO-JSON encodings. This is an area both NCO-JSON and CovJSON could address with the aim of increasing confidence in features and functionality.

3.3 The Appropriateness of JSON as a Container for EO Ranges

A common question which surfaced during the course of WG business was whether one JSON encoding improved over another with regards to representation of the actual array data values backing large coverages. Other common questions related to minimizing storage and network data volume, reducing client-side parsing time, and increasing human-readability. Unfortunately there is no simple answer to these questions however the WG deemed it of utmost importance to comment on how clients can mitigate against issues which can be associated with transferring and reading huge array data structures representing point observation values. The WG learned that the following mechanisms[2] were accepted to be of use when addressing this issue:

- **Encode the range values in a separate document(s) in a more efficient format:** CovJSON for example natively supports this feature. Various formats for the range are possible, but one attractive possibility is CovJSON itself, which provides a JSON encoding for a standalone multidimensional array (this can be compressed during data transfer for much greater efficiency). It may also, of course, be possible to use binary formats like NetCDF for this purpose. But many of these formats encode the full coverage (not just the range) so additional work is required in this area if these extensions were required.
- **Enable gzip compression**: if your server supports it, enable gzip for serving JSON responses. Especially for range data arrays this can dramatically reduce the resulting data volume and will lead to much lower loading times in clients.
- **Serve pre-gzipped static files**: if the JSON documents are not dynamically generated but instead stored on a static server, then, if your server supports that, consider transparently serving pre-gzipped files to reduce both server storage and processing power. This model may not always scale well with client access patterns but it can be useful if coverages are known before the fact.
- **Indent and order**: although indentation and field ordering is irrelevant for machine clients, it is important for humans. Therefore, use indentation and order fields.
- **Remove whitespace in data arrays**: the `values` arrays of range data can contain large numbers of elements. By default, many JSON serializers add whitespaces (including newlines) between elements. If gzip is used then the difference in data volume compared to having whitespace or not is minimal. However, after decompression, each of those whitespaces has to be parsed by the client, and this can lead to increased loading time. Therefore, if possible, the range arrays should have any whitespace omitted while serializing the JSON.
- **Avoid spurious digits**: when transforming from one data format to another it may happen that spurious digits get added to floating-point numbers. For example, a number may be stored on disk as exactly `0.01` (for example by storing it as an integer `1` together with a scaling factor `1e-2`). When reading this value back into a 32-bit floating point variable, the result is something like `0.0099999998`. Serializing this number in JSON would increases data volume and reduces compressibility because additional data in the form of spurious digits were added. Care should therefore be taken to avoid these issues if possible. The following strategies may help in doing that:
    - Use the floating-point data type with the highest precision available and hope for the best.
    - Apply rounding while serializing to JSON.
    - Read numbers into an arbitrary-precision decimal data type.

[2] Several of these were interpreted from the 'Encoding Advice' section of https://covjson.org/cookbook/#encoding-advice but have been generalized.

- **Investigate the use of tiling schemes for granular coverage representation:** Due to the ubiquity of map navigation as a means to ease the interpretation of large datasets in a recognizable manner, tiling schemes are important enough to warrant an entirely separate conversation in their own right. Clients accessing large coverage ranges will most likely wish to define the notion of 'zoom level(s)' which in turn determine the granularity of the data returned in the range response. It is unrealistic to expect clients such as Web browsers to load all coverage data at once for the higher zoom levels. Instead, the service serving the coverage should break up the data at each zoom level into a set of tiles, which are logically arranged in an order which the application understands. Then, for example, when a map scrolls to a new location, or to a new zoom level, the service determines which tiles are needed (using coordinates), and translates those values into a set of tiled ranges to retrieve. Although the purpose of this discussion is not to determine how best this should be done, it is worth mentioning some resources for consideration:
    - CartoDB's TorqueTiles [39] (based conceptually on TileMaps of Tile Map Service Specification [40])
    - GeoJSON-VT [41]
    - Google Map and Tile Coordinates [42]
    - Mapbox Vector Tile Specification [43]

  From the JSON encodings reviewed in this section, as of writing only CovJSON permits tiled arrays. Contextual information on this implementation exists at [44] [45] with the normative specification available at [46].
- **Investigate the use of hybrid JSON responses**; HDF5 v1.10.5 comes with an API that makes it possible to combine any JSON representation with the data payload stored in an HDF5 file as a single server response object. An HDF5 file can have so-called user block, which comes at the start of the file and has a known, discoverable, size. The actual HDF5 file content starts right after the user block. Any kind of content can be stored in the user block, for example, CovJSON (or NCO-JSON). Instead of the coverage range and domain values as text, the CovJSON would have byte offsets and lengths of the data's streams in the HDF5 payload. Any CovJSON client could retrieve CovJSON from the user block and then read in the actual data using the byte offset and size information stored there from the rest of the response body. An added benefit would be that saving that response to disk makes it a valid HDF5 file that can be later used if needed. Response sizes could benefit from having coverage data in binary format but applying HDF5 dataset compression would yield additional response size reduction.

The above offers various mechanisms for tackling inquiries which question to the appropriateness of JSON as a container for EO ranges. The WG accepts that this is not by any means an exhaustive list and anticipate numerous other mechanisms to be in existence. Again, the goal here is to stimulate discussion in this area with the aim of maturing JSON encodings for EO.

3.4 Ecosystem Maturity

The three encodings featured in Table 2, and more broadly, JSON encodings for representing EO coverages as a whole, represent a fairly young technology field with lots of exciting efforts, collaboration and technical discussions ongoing. At the time of writing, the most significant opportunity to mature this field generally is for them to be adopted by existing enterprise-grade EO software and used at that level. All three of the encodings listed in Table 2 meet this goal to some extent e.g. CovJSON infused into Hyrax, HDF5/JSON infused into HSDS, NCO-JSON in ERDDAP. However huge opportunity exists for each to extend infusion into other software projects e.g. any encoding infused into THREDDS.

3.5 Harmonization Efforts

Briefly, we consolidate several information resources peppered throughout the above narrative [3] [4] [5] [24] [36] [38].

4 Conclusion

The primary function of this paper is to ensure that the collective knowledge gained through a year ESDSWG *JSON Encodings in Earth Observation Coverages* Working Group is not lost and consequently that it can be considered, debated and hopefully utilized in other forums outside of NASA with the aim of driving progress in this field. The primary contribution of this paper is a technology review. That review addresses issues such as questioning the appropriateness of JSON as a container for Earth observation coverages, the ecosystem maturity and concludes with harmonization efforts. Ultimately, all of the JSON encodings featured in this discussion paper, and more broadly, JSON encodings for representing Earth observation coverages as a whole, represent a fairly young technology field with lots of exciting efforts, collaboration and technical discussions ongoing.

5 References

Informative References

[1] NASA Earth Science Data System Working Groups (2019) retrieved on 2019-03-17 from https://earthdata.nasa.gov/community/earth-science-data-system-working-groups-esdswg

[2] The JavaScript Object Notation (JSON) Data Interchange Format, ISSN 2070-1721, retrieved 2019-03-17 from https://tools.ietf.org/html/rfc8259

[3] Blower, J. Gallagher, J, Jelenak A. McGibbney, L. J. & Zender, C. (2018) JSON Encodings for Spatial Data: Data Modeling, Dialects and Languages, ESIP Summer 2018 Meeting, July 19th, 2018, Tucson, AZ, United States, retrieved on 2019-03-17 from https://sched.co/EyqZ

[4] Gallagher, J, Hemphill, C., LeBauer, D. McGibbney, L. J., Simons, B. & Zender, C. (2018) JSON Encodings for Spatial Data: Applications and Use Cases , ESIP Summer 2018 Meeting, July 19th, 2018, Tucson, AZ, United States, retrieved on 2019-03-17 from https://sched.co/Eyql

[5] Fils, D., Shepherd, A., Zender, C. S. & McGibbney, L. J. (2018) IN31B: JSON-Based Structured Data for Discovery, Access, and Usability: Leveraging Schema.org, eLightning

session, December 12th, 2018, Washington D.C., United States, retrieved on 2019-03-17 from https://agu.confex.com/agu/fm18/meetingapp.cgi/Session/57404

[6] EOSDIS Homepage (2019) retrieved on 2019-03-17 from https://earthdata.nasa.gov/about

[7] OGC Homepage (2019) retrieved on 2019-03-17 from http://www.opengeospatial.org/

[8] CF-JSON Homepage (2019) retrieved on 2019-03-17 from http://cf-json.org/specification

[9] CovJSON Homepage (2019) retrieved on 2019-03-17 from https://covjson.org/

[10] The GeoJSON Format (2016) Internet Engineering Task Force (IETF) Request for Comments: 7946, ISSN: 2070-1721, retrieved on 2019-03-17 from https://tools.ietf.org/html/rfc7946

[11] Specification and tools for representing HDF5 in JSON (2019) retrieved on 2019-03-17 from https://github.com/HDFGroup/hdf5-json

[12] HDF5/JSON Documentation (2019) retrieved on 2019-03-17 from http://hdf5-json.readthedocs.io/en/latest/

[13] JSON-LD 1.0 A JSON-based Serialization for Linked Data W3C Recommendation 16 January 2014, retrieved 2019-03-17 from https://www.w3.org/TR/json-ld/

[14] NCO 4.8.0-alpha04 User Guide (2019) retrieved on 2019-03-27 from http://nco.sourceforge.net/nco.html#json

[15] Feedback requested on NCO netCDF -> JSON converter (2016) retrieved on 2019-03-27 from http://www.unidata.ucar.edu/mailing_lists/archives/netcdfgroup/2016/msg00295.html

[16] NCO netCDF Operators Command-line operators for netCDF and HDF files (2016) retrieved on 2019-03-27 from https://sourceforge.net/p/nco/discussion/9829/thread/8c4d7e72/

[17] nc_json: JSON parser for netCDF, netCDF to JSON generator (2019) retrieved on 2019-03-27 from https://github.com/pedro-vicente/nc_json

[18] OGC (2019) OGC EO Dataset Metadata GeoJSON(-LD) Encoding Standard, retrieved on 2019-03-27 from https://portal.opengeospatial.org/files/?artifact_id=73344

[19] OGC (2016) OGC OpenSearch Extension for Earth Observation, 13-026r8, Version: 1.0.0, retrieved on 2019-03-27 from https://portal.opengeospatial.org/files/?artifact_id=71370&version=1

[20] OGC OWS Context GeoJSON Encoding Standard, OGC Document 14-055r2, 2017-04-06. Retreived 2019-03-27 from https://portal.opengeospatial.org/files/?artifact_id=68826

[21] STAR JSON (2019) retrieved on 2019-03-27 from http://www.space-research.org/blog/star_json.html

[22] star-json: A HDF5 JSON format specification and parser (2019) retrieved on 2019-03-27 from https://github.com/pedro-vicente/star-json

[23] topojson: An extension of GeoJSON that encodes topology! (2019) retrieved on 2019-03-27 from https://github.com/topojson/topojson

[24] CovJSON Specification: CovJSON, CF-JSON and NCO-JSON (2017) retrieved on 2019-03-27 from https://github.com/covjson/specification/issues/86

[25] The OGC Abstract Specification Topic 6: Schema for coverage geometry and functions, OGC Document 07-011, 2006-01-30. (Also available as ISO 19123) retrieved 2019-03-17 from http://portal.opengeospatial.org/files/?artifact_id=19820

[26] OGC testbed-14: Swath Coverage Engineering Report, publication date 2019-02-07, OGC 18-047r3 retrieved 2019-03-18 from http://docs.opengeospatial.org/per/18-047r3.html#SwathCoverage

[27] netCDF-LD Homepage (2019) retrieved on 2019-03-27 from https://binary-array-ld.github.io/netcdf-ld/

[29] science-on-schema.org Homepage (2019) retrieved on 2019-03-27 from https://github.com/ESIPFed/science-on-schema.org

[30] OGC Testbed-12 JSON and GeoJSON User Guide (2017) OGC reference: OGC 16-122r1, 2017-06-21retrieved on 2019-03-27 from http://docs.opengeospatial.org/guides/16-122r1.html#_json_for_coverages

[31] OGC From JSON to JSON-LD (2019) retrieved on 2019-03-27 from https://github.com/opengeospatial/architecture-dwg/blob/eefba7e2f250fd82b47ae4dc546525eb89eb1ecc/json-best-practice/clause_7-from_json_to_jsonld.adoc#json-for-coverages

[32] Overview of the CoverageJSON format (2019) W3C Editor's Draft 22 March 2019, retrieved on 2019-03-27 from http://w3c.github.io/sdw/coverage-json/

[33] OGC Coverages DWG Homepage (2019) retrieved on 2019-03-27 from http://www.opengeospatial.org/projects/groups/coveragesdwg

[34] OGC Web Feature Service 3.0: Part 1 – Core (2019) OGC internal reference 17-069, Version: 3.0.0-SNAPSHOT, retrieved on 2019-03-27 from http://docs.opengeospatial.org/DRAFTS/17-069.html

[35] OGC Web API Guidelines (2019) retrieved on 2019-03-27 from https://github.com/opengeospatial/OGC-Web-API-Guidelines

[36] Compare and contrast CoverageJSON with CF-NetCDF (2016) retrieved on 2019-03-27 from https://github.com/covjson/specification/issues/60

[37] OGC HDF SWG (2019) retrieved on 2019-03-27 from http://www.opengeospatial.org/projects/groups/hdfswg

[38] cf-json.github.io: Consolidation of `cf-json` and `nco-json` (2019) retrieved on 2019-03-27 from https://github.com/cf-json/cf-json.github.io/issues/10

[39] TorqueTiles 2.0 Specification (2015) retrieved on 2019-03-28 from https://github.com/CartoDB/torque-tiles/blob/master/2.0/spec.md

[40] Tile Map Service Specification v1.0 (2012) retrieved on 2019-03-28 from http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification

[41] Rendering big geodata on the fly with GeoJSON-VT (2015) retrieved on 2019-03-28 from https://blog.mapbox.com/rendering-big-geodata-on-the-fly-with-geojson-vt-4e4d2a5dd1f2

[42] Google Map and Tile Coordinates (2019) retrieved on 2019-03-28 from https://developers.google.com/maps/documentation/javascript/coordinates

[43] mapbox Vector Tile Specification v2.1 (2017) retrieved on 2019-03-28 from https://github.com/mapbox/vector-tile-spec/tree/master/2.1

[44] CovJSON tiling (2015) retrieved on 2019-03-28 from https://github.com/covjson/specification/issues/29

[45] Split up range into fragments? (2016) retrieved on 2019-03-28 from https://github.com/covjson/specification/issues/49

[46] CovJSON Specification – TiledNdArray Objects. Retrieved 2019-03-28 from https://covjson.org/spec/#tiledndarray-objects

6 Authors' Addresses

Dr. Lewis John McGibbney Ph.D., B.Sc.

Jet Propulsion Laboratory

California Institute of Technology

4800 Oak Grove Drive

Pasadena, California 91109-8099

Mail Stop : 158-256C

Email: lewis.j.mcgibbney@jpl.nasa.gov

The following individuals contributed to the WG: Justin Rive, Valerie Dixon, Aleksandar Jelenak, Charles Zender, Ted Habermann, James Gallagher, Corey Hemphill, David LeBauer, Bob Simons, Jon Blower, Mark Hedley, Doug Fils, Adam Shepherd, Gobe Hobona, Scott Simmons, Johannes Echteroff, Allan Doyle, Robert Ferarro, Keith Evans, Fan Fang, Jay Su, SiriJodha Khasla, Steve Olding, Tobias Kundig, Muqun Yang, Vardis Tsontos, Ed Armstrong, Michael Rilee, Doug Newman,

Appendix A – Glossary of Acronyms

Acronym	Description
DWG	Domain Working Group
EO	Earth Observation
EOSDIS	Earth Observing System Data and Information System
ESDS	Earth Science Data Systems

ESDSWG	Earth Science Data System Working Group
HSDS	Highly Scalable Data Server
JSON	JavaScript Object Notation
NASA	National Aeronautics and Space Administration
OGC	Open Geospatial Consortium
SWG	Standards Working Group
UMM-G	Unified Metadata Model for Granules
W3C	World Wide Web Consortium
WGISS	Working Group on Information Systems and Services